

What is claimed is:

1. A method of facilitating optimization processing in a compiler, comprising the steps of:

(a) storing, in a language-specific-rule table, one or more predetermined rules which are specified for one or more programming languages;

(b) analyzing a program code which includes one or more instructions, and is described in one of said one or more programming languages, based on said one or more predetermined rules, to obtain an analysis result; and

(c) embedding said analysis result in said program code.

2. A method according to claim 1, wherein in said step (a), said one or more predetermined rules are stored in said language-specific-rule table as one or more language-specific-information analyzing functions, and

said step (b) comprises the substeps of,

(d) reading out, from said language-specific-rule table, at least one of said one or more language-specific-information analyzing functions which is needed for analyzing said program code, and

(e) determining values of or relationships between variables included in said program code, based on said at least one of said one or more language-

specific-information analyzing functions read out in said step (d), and producing said analysis result which includes the determined values of or relationships between the variables.

5

3. A method according to claim 2, wherein said step (b) further comprises the substep of,

(f) said at least one of said one or more language-specific-information analyzing functions read out in said step (d) is registered in a check function table for use in said step (e).

4. A method according to claim 3, wherein the operation in step (b) is performed for each instruction set which is comprised of at least one of said one or more instructions, and

in said step (c), said analysis result is embedded in a position preceding said each instruction set in said program code.

20

5. A method according to claim 1, wherein said program code is a source code.

6. A method according to claim 1, wherein said program code is an intermediate code.

7. An apparatus for facilitating optimization

processing in a compiler, comprising:

a language-specific-rule table which stores one or more predetermined rules which are specified for one or more programming languages;

5 an analyzing unit which analyzes a program code which includes one or more instructions, and is described in one of said one or more programming languages, based on said one or more predetermined rules, to obtain an analysis result; and

10 an embedding unit which embeds said analysis result in said program code.

8. An apparatus according to claim 7, wherein said language-specific-rule table stores said one or more predetermined rules as one or more language-specific-information analyzing functions, and

said analyzing unit comprises,

20 a readout unit which reads out, from said language-specific-rule table, at least one of said one or more language-specific-information analyzing functions which is needed for analyzing said program code, and

25 a determination unit which determines values of or relationships between variables included in said program code, based on said at least one of said one or more language-specific-information analyzing functions read out by said readout unit, and produces said

analysis result which includes the determined values of or relationships between the variables.

9. An apparatus according to claim 8, wherein
5 said analyzing unit comprises,

a check function table in which said at least one of said one or more language-specific-information analyzing functions read out by said readout unit is registered for use by said determination unit.

10

10. An apparatus according to claim 9, wherein the operation of said analyzing unit is performed for each instruction set which is comprised of at least one of said one or more instructions, and

15

said embedding unit embeds said analysis result in a position preceding said each instruction set in said program code.

11. An apparatus according to claim 7, wherein
20 said program code is a source code.

12. An apparatus according to claim 7, wherein said program code is an intermediate code.

13. A product for use with an apparatus for
25 facilitating optimization processing in a compiler, said product, when used with said apparatus, is able to

output control information which directs the apparatus to comprise:

a language-specific-rule table which stores one or more predetermined rules which are specified for
5 one or more programming languages;

an analyzing unit which analyzes a program code which includes one or more instructions, and is described in one of said one or more programming languages, based on said one or more predetermined rules,
10 to obtain an analysis result; and

an embedding unit which embeds said analysis result in said program code.

14. A product according to claim 13, wherein said
15 language-specific-rule table stores said one or more predetermined rules as one or more language-specific-information analyzing functions, and

said analyzing unit comprises,

a readout unit which reads out, from said
20 language-specific-rule table, at least one of said one or more language-specific-information analyzing functions which is needed for analyzing said program code, and

a determination unit which determines values
25 of or relationships between variables included in said program code, based on said at least one of said one or more language-specific-information analyzing functions

read out by said readout unit, and produces said analysis result which includes the determined values of or relationships between the variables.

5 15. A product according to claim 13, wherein said program code is a source code.

10 16. A product according to claim 13, wherein said program code is an intermediate code.

15 17. A compiler comprising:
 a front end unit which performs syntax analysis of a source code which is described in one of one or more predetermined programming languages, to produce an intermediate code;

 a language-specific-rule table which stores one or more predetermined rules which are specified for said one or more predetermined programming languages;

20 an analyzing unit which analyzes said intermediate code which includes one or more instructions, based on said one or more predetermined rules, to obtain an analysis result;

25 an embedding unit which embeds said analysis result in said program code to produce a modified intermediate code; and

 an optimizing unit which performs an operation of optimizing said modified intermediate code.

18. A compiler according to claim 17, wherein
said language-specific-rule table stores said one or
more predetermined rules as one or more language-
5 specific-information analyzing functions, and

said analyzing unit comprises,

a readout unit which reads out, from said
language-specific-rule table, at least one of said one
or more language-specific-information analyzing
10 functions which is needed for analyzing said
intermediate code, and

a determination unit which determines values
of or relationships between variables included in said
intermediate code, based on said at least one of said
15 one or more language-specific-information analyzing
functions read out by said readout unit, and produces
said analysis result which includes the determined
values of or relationships between the variables.

20 19. A compiler comprising:

a language-specific-rule table which stores
one or more predetermined rules which are specified for
one or more programming languages;

an analyzing unit which analyzes a source
25 code which includes one or more instructions, and is
described in one of said one or more programming
languages, based on said one or more predetermined rules,

to obtain an analysis result;

an embedding unit which embeds said analysis result in said source code to produce a modified source code;

5 a syntax analyzing unit which performs syntax analysis of said modified source code which is described in one of one or more predetermined programming languages, to produce an intermediate code; and

10 an optimizing unit which performs an optimization operation on said intermediate code.

20. A compiler according to claim 19, wherein said language-specific-rule table stores said one or
15 more predetermined rules as one or more language-specific-information analyzing functions, and

said analyzing unit comprises,

a readout unit which reads out, from said language-specific-rule table, at least one of said one
20 or more language-specific-information analyzing functions which is needed for analyzing said source code, and

a determination unit which determines values of or relationships between variables included in said
25 source code, based on said at least one of said one or more language-specific-information analyzing functions read out by said readout unit, and produces said

analysis result which includes the determined values of
or relationships between the variables.

03046799 074400